

09/500,921

1. (Currently Amended) A method for processing ~~a one-to-one~~ the same request from a client program to multiple instances of a the same server program over a the same protocol, said method comprising:
  - transferring said same request from said client program to ~~a multiplexer~~ an intermediary;
  - generating a plurality of ~~protocol~~ instances of said same request using said ~~multiplexer intermediary~~, wherein each of said ~~protocol~~ instances of said same request corresponds to a different instance of said same server program;
  - transferring said ~~protocol~~ instances of said same request from said ~~multiplexer intermediary~~ to said instances of said same server program;
  - transferring a plurality of responses from said instances of said same server program to said ~~multiplexer intermediary~~;
  - converting said responses to a uniform response; and
  - transferring said uniform response to said client program.
2. (Currently Amended) The method in claim 1, further comprising specifying target instances of said same server program to form a fan out target list, to which said ~~protocol~~ instances of said same request will be transferred.
3. (Original) The method in claim 1, wherein said converting comprises selecting an operation to combine said responses.
4. (Original) The method in claim 3, wherein said operation comprises one of listing said responses, aggregating said responses, adding said responses, preparing a subset of said responses, identifying a maximum of said responses, identifying a minimum of said responses, and averaging said responses.
5. (Currently Amended) The method in claim 1, wherein said ~~multiplexer intermediary~~ automatically creates said ~~protocol~~ instances of said same request.

09/500,921

6. (Currently Amended) The method in claim 1, wherein said client program, ~~said instances of said same server program~~, and said same protocol are not modified by said method.
7. (Original) The method in claim 1, wherein said unified response has an instance corresponding to said client program.
8. (Currently Amended) A method of processing a the same request from a client program to multiple instances of a the same server program over a the same protocol, said method comprising:
  - modifying said same request to create multiple instances of said same request, each of said ~~protocol~~ instances of said same request corresponding to a single instance of said same server program;
  - transferring said ~~protocol~~ instances of said same request to corresponding ones of said instances of said same server program; and
  - modifying and combining responses to said same request from said instances of said same server program to create a unified response.
9. (Original) The method in claim 8, wherein a ~~multiplexer~~ an intermediary alters said same request to comply with each instance of said same server program.
10. (Currently Amended) The method in claim 9, wherein said ~~multiplexer~~ intermediary automatically creates said ~~protocol~~ instances of said same request.
11. (Currently Amended) The method in claim 8, further comprising specifying target instances of said same server program to form a fan out target list, to which said ~~protocol~~ instances of said same request will be transferred.
12. (Original) The method in claim 8, wherein said converting comprises selecting an operation to combine said responses.

09/500,921

13. (Original) The method in claim 12, wherein said operation comprises one of listing said responses, aggregating said responses, adding said responses, preparing a subset of said responses, identifying a maximum of said responses, identifying a minimum of said responses, and averaging said responses.
14. (Currently Amended) The method in claim 8, wherein said client program, said ~~instances of~~ said same server program, and said same protocol are not modified by said method.
15. (Original) The method in claim 8, wherein said unified response has an instance corresponding to said client program.
16. (Currently Amended) A method of using a computer program to process ~~a one-to-one~~ the same request from a client program to multiple instances of a the same server program over a the same protocol, said method comprising:
- using said computer program to transfer said same request from said client program to ~~a multiplexer or an intermediary~~;
  - using said computer program to generate a plurality of ~~protocol~~ instances of said same request using said ~~multiplexer-intermediary~~, wherein each of said ~~protocol~~ instances of said same request corresponds to a different instance of said same server program;
  - using said computer program to transfer said ~~protocol~~ instances of said same request from said ~~multiplexer-intermediary~~ to said instances of said same server program;
  - using said computer program to transfer a plurality of responses from said instances of said same server program to said ~~multiplexer-intermediary~~;
  - using said computer program to convert said responses to a uniform response; and
  - using said computer program to transfer said uniform response to said client program.
17. (Currently Amended) The method in claim 16, further comprising using said computer program to specify target instances of said same server program to form a fan out target list, to which said same request will be transferred.

09/500,921

18. (Original) The method in claim 16, wherein said using said computer program to convert comprises using said computer program to select an operation to combine said responses.
19. (Original) The method in claim 18, wherein said operation comprises one of listing said responses, aggregating said responses, adding said responses, preparing a subset of said responses, identifying a maximum of said responses, identifying a minimum of said responses, and averaging said responses.
20. (Currently Amended) The method in claim 16, wherein said ~~multiplexer~~ intermediary automatically creates said ~~protocol~~ instances of said same request.
21. (Currently Amended) The method in claim 16, wherein said client program, ~~said instances of~~ said instances of said same server program, and said same protocol are not modified by said computer program.
22. (Original) The method in claim 16, wherein said unified response has an instance corresponding to said client program.
23. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform a method for processing a ~~one-to-one~~ the same request from a client program to multiple instances of a the same server program over a the same protocol, said method comprising:
- transferring said same request from said client program to a ~~multiplexer~~ an intermediary;
  - generating a plurality of ~~protocol~~ instances of said same request using said ~~multiplexer-intermediary~~, wherein each of said ~~protocol~~ instances of said same request corresponds to a different instance of said same server program;
  - transferring said ~~protocol~~ instances of said same request from said ~~multiplexer intermediary~~ to said instances of said same server program;

09/500,921

transferring a plurality of responses from said instances of said same server program to said ~~multiplexer-intermediary~~;

converting said responses to a uniform response; and

transferring said uniform response to said client program.

24. (Currently Amended) The program storage device in claim 23, further comprising specifying target instances of said same server program to form a fan out target list, to which said ~~protocol~~ instances of said request will be transferred.
25. (Original) The program storage device in claim 23, wherein said converting comprises selecting an operation to combine said responses.
26. (Original) The program storage device in claim 25, wherein said operation comprises one of listing said responses, aggregating said responses, adding said responses, preparing a subset of said responses, identifying a maximum of said responses, identifying a minimum of said responses, and averaging said responses.
27. (Currently Amended) The program storage device in claim 23, wherein said ~~multiplexer-intermediary~~ automatically creates said ~~protocol~~ instances of said same request.
28. (Currently Amended) The program storage device in claim 23, wherein said client program, ~~said instances of~~ said instances of said same server program, and said same protocol are not modified by said method.
29. (Original) The program storage device in claim 23, wherein said unified response has an instance corresponding to said client program.
30. (Currently Amended) A ~~multiplexer~~ An intermediary for processing a ~~one-to-one~~ the same request from a client program to multiple instances of a the same server program over a the same protocol, said ~~multiplexer-intermediary~~ comprising:

09/500,921

a converter for generating a plurality of ~~protocol~~ instances of said same request, wherein each of said ~~protocol~~ instances of said same request corresponds to a different instance of said same server program; and

a response combiner for converting said responses to a uniform response.

31. (Currently Amended) The ~~multiplexer-intermediary~~ in claim 30, wherein said response combiner selects an operation to combine said responses.
32. (Currently Amended) The ~~multiplexer-intermediary~~ in claim 31, wherein said operation comprises one of listing said responses, aggregating said responses, adding said responses, preparing a subset of said responses, identifying a maximum of said responses, identifying a minimum of said responses, and averaging said responses.
33. (Currently Amended) The ~~multiplexer-intermediary~~ in claim 30, wherein said converter automatically creates said ~~protocol~~ instances of said same request upon receipt of said same request.
34. (Currently Amended) The ~~multiplexer-intermediary~~ in claim 30, wherein said client program, ~~said instances of said~~ same server program, and said same protocol are not modified by said ~~multiplexer-intermediary~~.
35. (Currently Amended) The ~~multiplexer-intermediary~~ in claim 30, wherein said unified response has an instance corresponding to said client program.